

REMARKS

Claims 1-17 are remaining in the present patent application. Claims 1, 10 and 15 have been amended. No new matter has been added. Support for the amendments to Claims 1, 10 and 15 can be found, among other places, in the instant application at original Claim 9 and lines 11-18 of page 6, which states, Service processor 120 is integrated as an input/output (I/O) device to server 110, and acts as an autonomous embedded device, which is powered independently and runs embedded applications independent of server 110's state. Server 110 may properly function with or without service processor 120 or with service processor 120 being inoperative. Further, service processor 120 is commercially available without a terminal, and is referred to as an embedded management processor or device because service processor 120 is part of server 110 and provides management services for server 110 (emphasis added).

35 U.S.C. §102 Rejection

In paragraph 7, the Office Action rejected Claims 1-17 under 35 U.S.C. 102(b) as being unpatentable over U.S. patent publication no. 2002/0029299 by Kappel et al. (referred to hereinafter as Kappel). Applicants have reviewed the above cited reference and respectfully submit that the present invention, as recited in Claims 1-17, is neither anticipated nor rendered obvious by the Kappel et al. reference.

Claim 1 recites,

An exception handling mechanism comprising:

an exception handler for recording exception information dependant on types of exceptions and programming tasks that encounter exceptions; and

a recovery agent for taking an action upon an occurrence of an exception that occurred for a programming task, wherein the action is performed outside of a debugging operation;

wherein the action to be taken upon the occurrence of the exception corresponds to a type of exception and a programming task, and includes one or a combination of restarting the programming task, terminating the programming task, resetting a system running the programming task, and disregarding the exception,

wherein the exception handler and the recovery agent run on a first system operates autonomously and is embedded in a second system.

Applicants respectfully submit that Kappel does not teach or suggest, among other things "...one or a combination of restarting the programming task, terminating the programming task, resetting a system running the programming task, and disregarding the exception, wherein the exception handler and the recovery agent run on a first system operates autonomously and is embedded in a second system."

According to the Federal Circuit, "[a]nticipation requires the disclosure in a single prior art reference of each claim under consideration" (W.L. Gore & Assocs. v. Garlock Inc., 721 F.2d 1540, 220 USPQ 303, 313 (Fed. Cir. 1983)). However, it is not sufficient that the reference recite all the claimed elements. As stated by the Federal Circuit, the prior art reference must disclose each element of the claimed invention "arranged as in the claims" (emphasis added; Lindermann Maschinenfabrik GmbH v. American Hoist & Derrick Co., 730 F.2d 1452, 221 USPQ 481, 485 (Fed. Cir. 1984)).

Kappel teaches a method for exception handling. Referring to the abstract, "the system includes a class creator that establishes a plurality of classes of exception types and an exception capture mechanism that captures an exception." Referring to the last sentence of paragraph 0003, Kappel's goal is to "...provide for a generalized or simplistic view of exception or error handling." Referring to the first sentence of paragraph 0029, Kappel achieves his goal by utilizing three generic exception types. Kappel states in the second to last sentence of paragraph 0029 that the exception types are application exceptions, system exceptions and validation exceptions.

At lines 6 and 7 of paragraph 0015 Kappel states, "...the exception handling system 50 is implemented in software, as an executable program..." In the last 3 sentences of paragraph 0019, Kappel states, "In the example of FIG.

2, the software in the memory 22 includes the exception handling system 50 and a suitable operating system (O/S) 26.” In lines 2-3 of paragraph 0016, Kappel states, “the computers 11 or 14 include a processor 21, memory 22...” Therefore, Kappel’s exception handling system 50 is software that resides in memory 22 of computers 11 or 14.

The Office Action asserts that Kappel teaches “one or a combination of restarting the programming task, terminating the programming task, resetting a system running the programming task, and disregarding the exception” at steps 58 and 59 depicted in Figure 3 and related text. The only place that Kappel refers to step 58 is in paragraph 0032. Paragraph 0032 states,

However, if it is determined at step 54 that the exception experience is not a validation exception, then the exception handling system 50 of the present invention propagates the exception to a central place for handling at step 58.

Note that propagating an exception to a central place for handling does not teach or suggest, “one or a combination of restarting the programming task, terminating the programming task, resetting a system running the programming task, and disregarding the exception.”

The only places that Kappel refers to step 59 is in the second sentence of paragraph 0032 and the last 2 lines of paragraph 0031. The second sentence of paragraph 0032 states, “The exception handling system 50 then exits at step 59.” The last 2 lines of paragraph 0031 states, “the present invention proceeds to step 55 to continue execution of the current module and then exits at step 59.” The last 2 lines of paragraph 0031 cannot be used to teach “one or a combination of restarting the programming task, terminating the programming task, resetting a system running the programming task, and disregarding the exception.”

Note that the text related to step 59 is silent with respect to “one or a combination of restarting the programming task, terminating the programming task, resetting a system running the programming task, and disregarding the exception.” Further, based on Applicants’ review of Kappel, Kappel does not teach or suggest “restarting the programming task, terminating the programming task, resetting a system running the programming task, and disregarding the exception” anywhere.

In the response to arguments section, the Office Action states “...Kappel in Fig. 3, step 59, ‘EXIT EXCEPTION HANDLING SYSTEM’, discloses a method for terminating the programming task.” Step 59 says “exit exception handling system.” An exception handling system can be exited without terminating the task that caused the exception handling system to be entered due to an exception associated with the task. For example, the last 4 lines of paragraph 0031 states, “If it is determined at step 54 that the exception experience is a validation exception, the exception handling system 50 of the present invention proceeds to step 55 to continue execution of the current module and then exits at step 59” (emphasis added).

Independent Claim 1 was amended to recite features associated with original Claim 9. The Office Action asserted that Kappel teaches the features cited by original Claim 9 at Figure 1. However, as already described herein, Kappel’s exception handling system 50 is software that resides in memory 22 of computers 11 or 14. Therefore, Kappel does not teach or suggest, “wherein the exception handler and the recovery agent run on a first system, that operates autonomously, embedded in a second system,” as recited by Claim 1.

Therefore, for at least these reasons independent Claim 1 should be patentable over Kappel. Further, independent Claims 10 and 15 should be patentable over Kappel for similar reasons that Claim 1 should be patentable

over Kappel. Kappel does not teach all of the elements as recited by Claims 1, 10 and 15.

In addition, Claim 15 recites “an exception handler for recording exception information on non-volatile memory...” The Office Action asserts that Kappel's item 50 teaches “an exception handler for recording exception information on non-volatile memory...” Item 50 is Kappel's exception handling system. However, note that Claim 15 recites “...information on non-volatile memory.” An exception handling system is an executable. It is not information. Further, no where does Kappel teach that his memory 22 is non-volatile.

Claims 2-9 depend on Claim 1. Claims 11-14 depend on Claim 10. Claims 16-17 depend on Claim 15. These dependent claims include all of the limitations which their respective independent claims recite. Further, these dependent claims include additional limitations which further make them patentable.

For example, Claim 2 recites “an exception is associated with a signature for identifying the recorded exception information with its associated exception.” The Office Action asserts that Kappel teaches Claim 2 at paragraph 0029 on page 3. Although paragraph 0029 includes the word “signature” the signature referred to by paragraph 0029 is a “method signature” not “a signature for identifying the recorded exception information with its associated exception,” as recited by Claim 2.

In a second example, Claim 4 recites, “each set of exception information being associated with a signature for identifying that set of exception information.” The Office Action asserted that Kappel teaches the embodiment recited by Claim 14 at “example 3, [0030].” Paragraph 0030 says nothing about signatures. Paragraph 0030 does mention an exception type, however, Kappel's

exception type is not a signature nor is it used for identifying a set of exception information.

In a third example, Claim 6 recites “wherein the recorded exception information includes data related to a program stack...” (emphasis added). The Office Action asserts that Kappel teaches the embodiment recited by Claim 6 at paragraph 0027. Paragraph 0027 states, “Thrown exceptions are propagated through the lexical block structure of a method, and then up through the method stack.” However, paragraph 0027 does not state that he includes data related to a program stack in recorded exception information.

In a fourth example, Claim 8 recites, “wherein the analysis tool uses a version to match the object code of a program running the programming task to the source code of the program.” The Office Action asserted that Kappel teaches the embodiment recited by Claim 8 at step 52 on page 3. Step 52 states, “identifying exception type using exception dictionary.” Kappel’s exception type is not a version nor is it used to match object code of a program to source code of the program.

In a fifth example, Claim 12 recites “wherein the exception information stored in the nonvolatile memory is compressed.” The Office Action asserts that 22 in Figure 2 teaches the embodiment recited by Claim 12. However, 22 refers to memory. Kappel does not teach that the memory is nonvolatile memory nor does Kappel teach that exception information is compressed.

Therefore the dependent claims should be patentable for at least the reasons that their respective independent claims should be patentable.

CONCLUSION

In light of the facts and arguments presented herein, Applicants respectfully request reconsideration of the rejected Claims.

Based on the arguments presented above, Applicants respectfully assert that Claims 1-17 overcome the rejections of record. Therefore, Applicants respectfully solicit allowance of these Claims.

The Examiner is invited to contact Applicants' undersigned representative if the Examiner believes such action would expedite resolution of the present Application.

Respectfully submitted,
Wagner Blecher LLP

Date: 10/26/2007



John P. Wagner Jr.
Registration No. 35,398

Address: Westridge Business Park
123 Westridge Drive
Watsonville, California 95076 USA

Telephone: (408) 377-0500 Voice
(408) 234-3649 Direct/Cell
(831) 722-2350 Facsimile